REMARKS

Status of the Claims

Claims 1, 3-8, 10-14, and 16-20 are currently present in the Application, and claims 1, 8, and 14 are independent claims. Claims 1, 8, and 14 have been amended, no claims have been canceled, and no claims have been added.

Applicants are not conceding that the subject matter encompassed by claims 1-20, prior to this and previous amendments, are not patentable over the art cited by the Examiner. Claims 1, 8, and 14 were amended in this Amendment solely to facilitate expeditious prosecution of this Application. Applicants respectfully reserve the right to pursue claims, including the subject matter encompassed by claims 1-20 as presented prior to this and previous Amendments, and additional claims in one or more continuing applications.

Examiner Interview

Applicants note with appreciation the telephonic interview conducted between Applicants' representative and the Examiner on May 19, 2008. During the telephonic interview, the Examiner and Applicants' representative discussed one of the 103 references (Oram, et al., "Managing Projects with Make"). In particular, Applicants' representative discussed that Applicants' invention retrieves source code subtasks included in a single source code file, selects a particular processor type for each subtask, and creates processor-type specific object code for each of the subtasks, which results in a single object file that includes multiple object code subtasks. In contrast, Oram discloses the creation of a library file, which packages two already generated object files. Applicants' representative asserted that Oram's library file is different than Applicants' object file and suggested amending Applicants' independent claims to distinctly claim that the object file is the direct result of compiling source code. No agreement was reached regarding the claims.

Claim Rejections - Alleged Anticipation Under 35 U.S.C. § 103

Claims 1, 3, 6-8, 10, 13-14, 16, and 19-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Oram, et al., ("Managing Projects with Make," hereinafter "Oram") in view of Stallman ("Using and Porting the GNU Compiler Collection for GCC 3.1," hereinafter "Stallman"). Applicants respectfully traverse these rejections.

Applicants have amended independent claim 1 to distinctly claim that Applicants' object code is a direct result of compiling source code. Support for such amendments can be found in Applicants' specification on page 45, lines 3-4 and, therefore, no new matter is added with such amendments. As amended, independent claim 1 is directed to a method for compiling source code for a plurality of heterogeneous processors with limitations comprising:

- receiving source code that includes a plurality of source code subtasks;
- independently selecting a processor type from the plurality of heterogeneous processor types for each of the plurality of source code subtasks, the independent selection comprising:
    - selecting a first processor type from the plurality of heterogeneous processor types for a first source code subtask included in the source code; and
    - selecting a second processor type from the plurality of heterogeneous processor types for a second source code subtask included in the source code, wherein the second processor type is different than the first processor type; and
- after the processor type selections, compiling the source code, which directly creates an object file that includes a first object code corresponding to the first source code subtask and a second object code corresponding to the second source code subtask, wherein the first object code is adapted to be processed by the first processor type and the second object code is adapted to be processed by the second processor type.

Applicants individually retrieve source code subtasks included in a source code file, select a particular processor type for each subtask, and create processor-type specific object code for each of the subtasks, which results in a single object file that

includes multiple object code subtasks. In contrast, Oram discloses the use of C programming "make" commands to selectively compile source code into an object code file. Oram's make command, however, is designed to produce an object file for **each** source code file. Oram states:

> "To compile the statements you want, define the proper symbols in CFLAGS. For instance, the following *make* command compiles ***all*** source files with the BSD and STATS symbols defined:" (page 79, emphasis added)

As can be seen from the above excerpt, Oram's "make" command generates a single object file from source code. The Office Action points to an excerpt in Oram to reject Applicants' last limitation but, after further review of Oram, Oram does not teach such limitation. On page 33, Oram states:

> "Compilation produces two object files, each containing several modules. Some compilers give each module the exact same name as the C function, while some add a leading underscore....**You then create a library, which we will call libops, from the two object files**...(You might also have to run ranlib(1) or some other utility to sort the library, depending on your system and your application)." (emphasis added)

As can be seen from the above excerpt, Oram's library file is created from two already created object files, and in no way is the same as "*compiling the source code, which directly creates an object file*" as claimed by Applicants. Furthermore, Applicants' invention allows a user to go from source code directly to an object file that includes multiple object code subtasks. Oram requires a user to perform an extra step because a user first creates object files, and then creates the library file. Therefore, not only is Oram's library file different than Applicants' object file, Oram requires a user to perform more steps to create the library file. The Office Action does not suggest that Stallman teaches or suggests such limitations and, indeed, Stallman does not teach such limitation.

Therefore, since neither Oram nor Stallman teach or suggest, either alone or in combination with each other, all the limitations included in Applicants' claim 1 as amended, amended claim 1 is allowable over Oram in view of Stallman. Claim 8 is an

information handling claim including similar limitations as claim 1 and, therefore, is allowable over Oram in view of Stallman for at least the same reasons that claim 1 is allowable as discussed above. Claim 14 is computer program product claim including similar limitations as claim 1 and, therefore, is allowable over Oram in view of Stallman for at least the same reasons that claim 1 is allowable as discussed above.

Each of the remaining claims 3, 6-7, 10, 13, 16, and 19-20 depend, either directly or indirectly, upon one of the allowable independent claims 1, 8 or 14. Therefore, each of claims 3, 6-7, 10, 13, 16, and 19-20 are allowable for at least the same reasons as there respective independent claims are allowable as discussed above.

Claims 4, 5, 11, 12, 17, and 18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Oram in view of Stallman in further view of Per Bothner ("Compiling Java with GCJ," hereinafter "Per Bothner"). Applicants respectfully traverse these rejections.

Each of claims 4, 5, 11, 12, 17, and 18 depend, either directly or indirectly, upon one of the allowable independent claims 1, 8, or 14. Therefore, each of claims 4, 5, 11, 12, 17, and 18 is allowable for at least the same reasons as there respective independent claims are allowable as discussed above.

Conclusion

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By /Leslie A. Van Leeuwen, Reg. No. 42,196/

Leslie A. Van Leeuwen, Reg. No. 42,196
Van Leeuwen & Van Leeuwen
Attorney for Applicants
Telephone: (512) 301-6738
Facsimile: (512) 301-6742